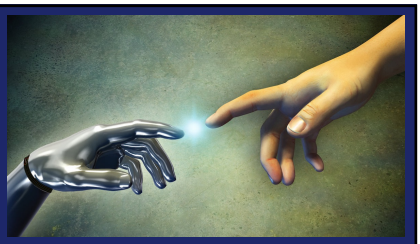
Introduction to Artificial Intelligence



DANIEL RACOCEANU
PROFESOR
SORBONNE UNIVERSITY
OCTOBER 2020
DANIEL.RACOCEANU@SORBONNE-UNIVERSITE.FR

SORBONNE UNIVERSITÉ

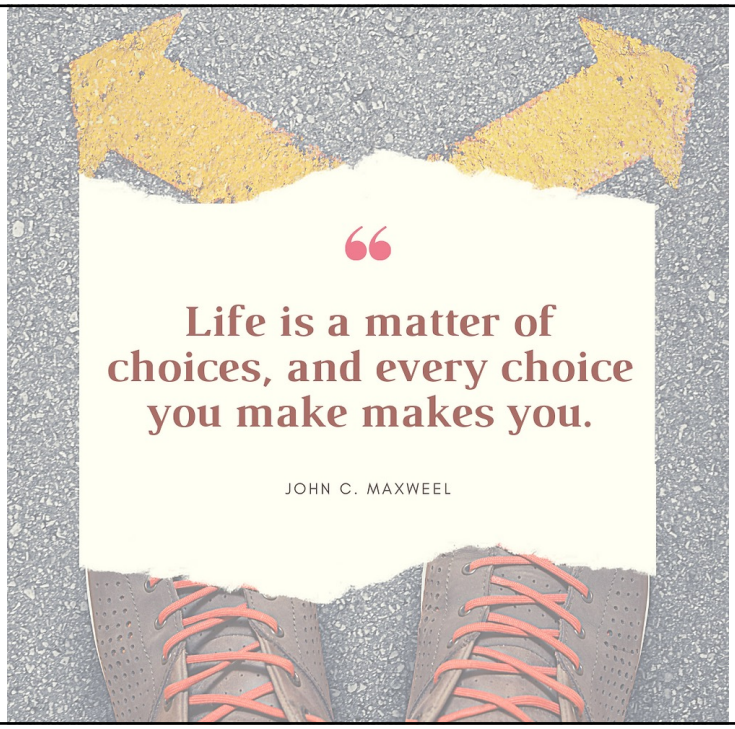
1

Arbres de Décision
Decision Trees

DANIEL RACOCEANU
PROFESOR,
SORBONNE UNIVERSITY
OCTOBER 2020
DANIEL.RACOCEANU@SORBONNE-UNIVERSITE.FR

SORBONNE UNIVERSITÉ

3

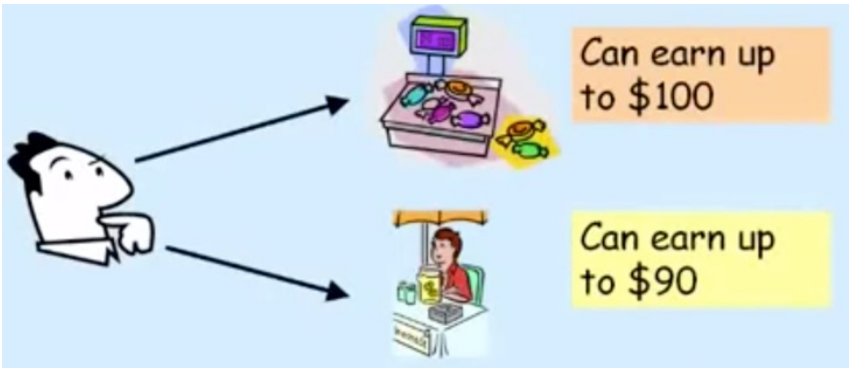


4

4

Choice between two business projects

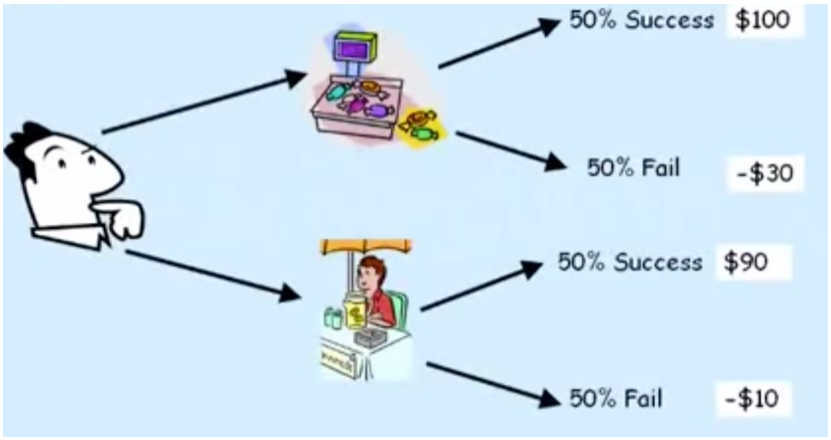
Which one will you chose ?



5

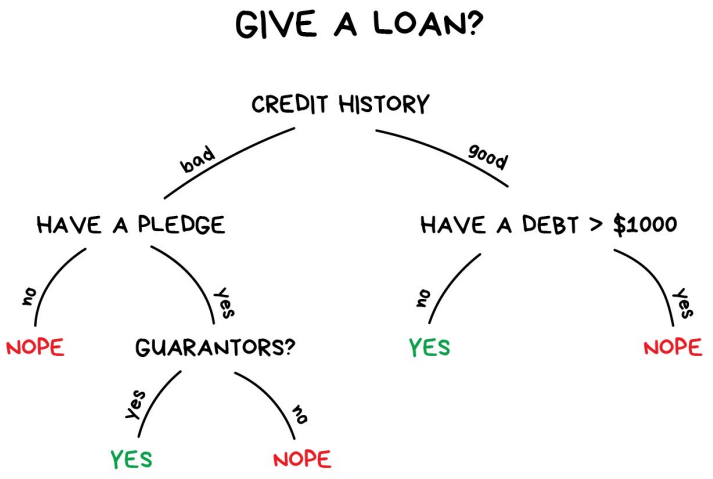
5

How about now ?



6

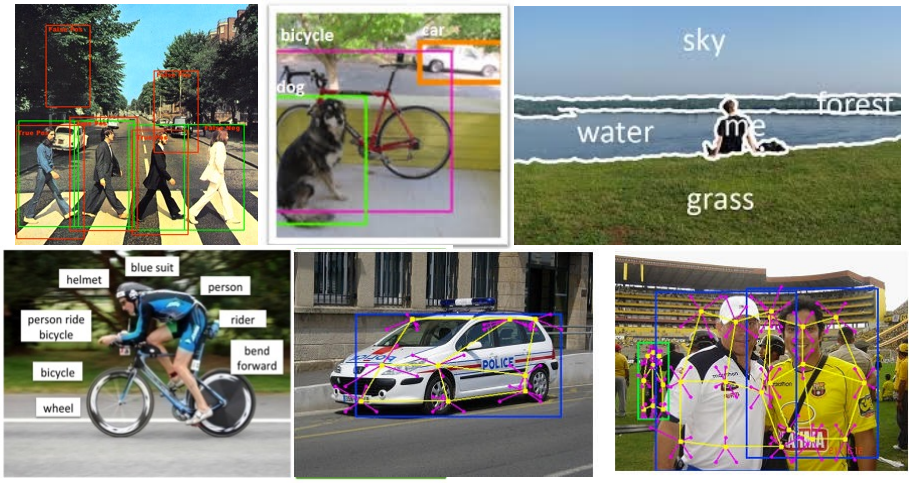
Accorder un prêt ...
Offering a loan ...



DECISION TREE

7

Motivation: Object Detection

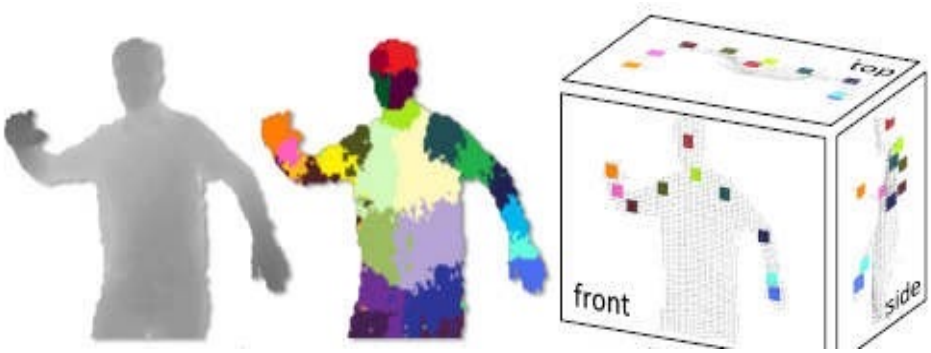


SORBONNE UNIVERSITÉ

Motivation: Kinect

SORBONNE UNIVERSITÉ

Squelettisation de personnes avec Kinect

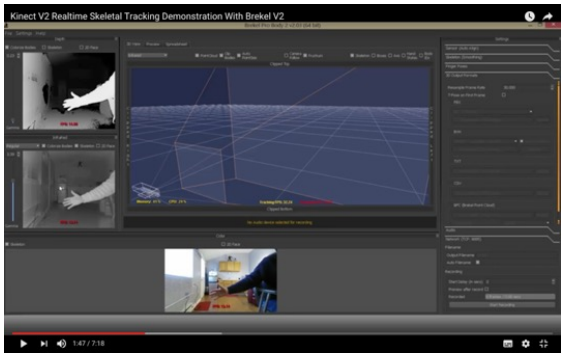


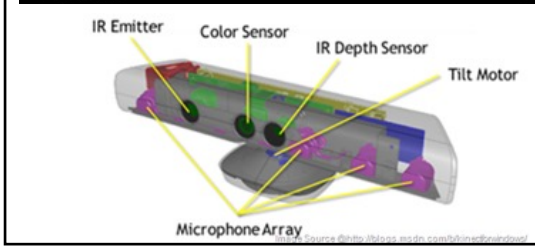
depth image ➡ body parts ➡ 3D joint proposals

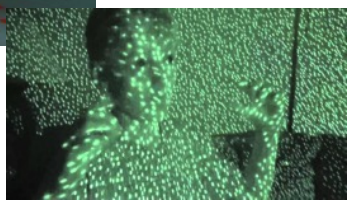
SORBONNE UNIVERSITÉ

10

Motivation: Kinect



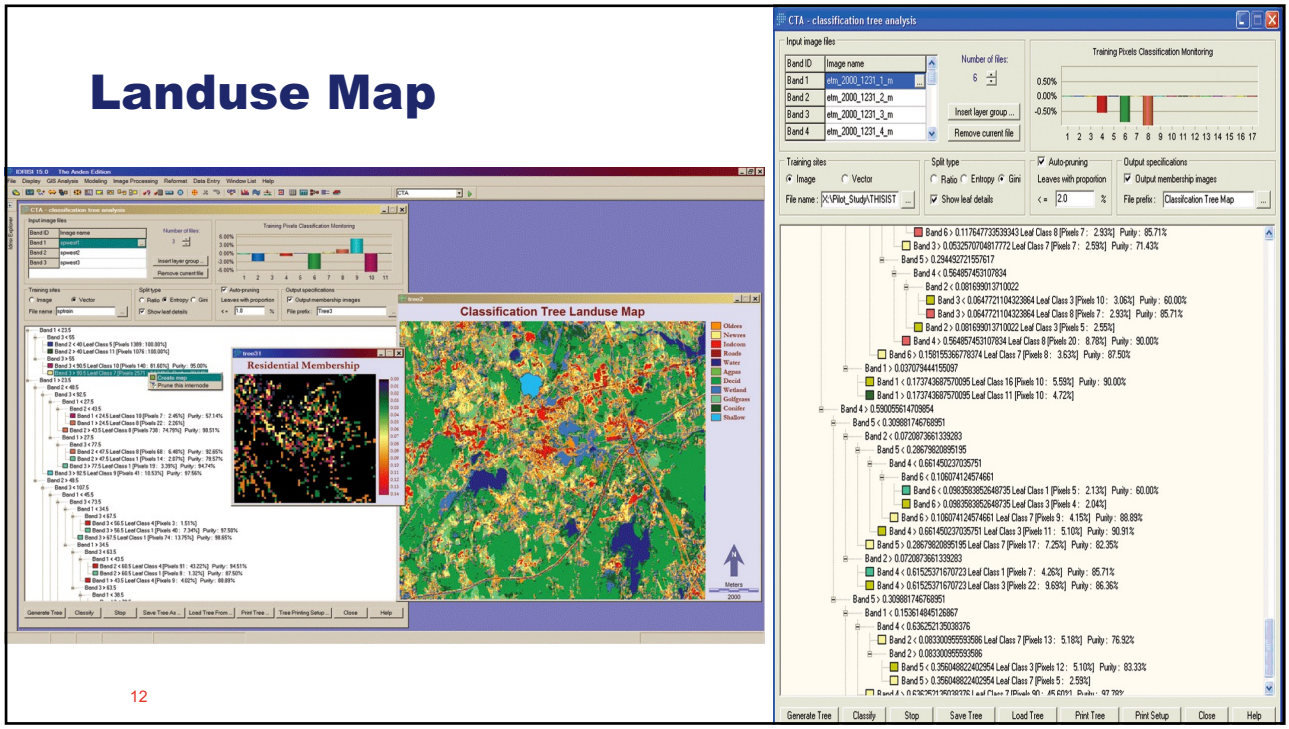




SORBONNE UNIVERSITÉ

11

Landuse Map



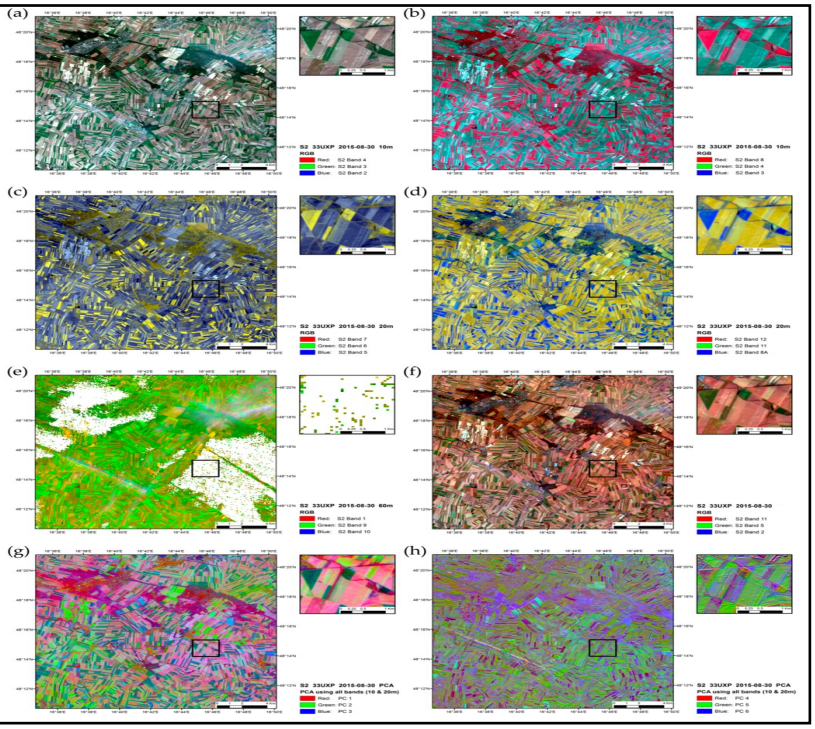
12

12

Télédétection
Remote sensing

Classifications des espèces d'arbres en Europe centrale

Tree Species Classifications in Central Europe



13

13

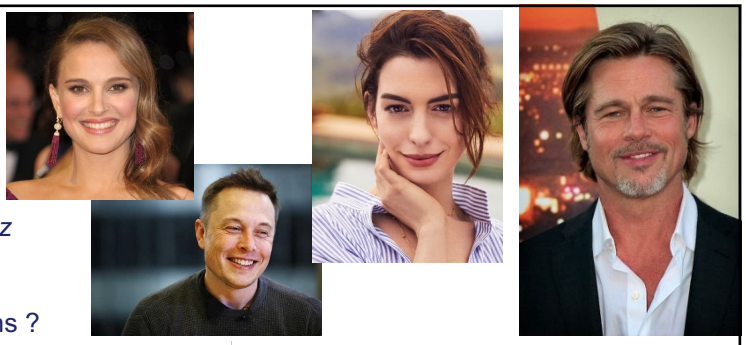
20 Questions

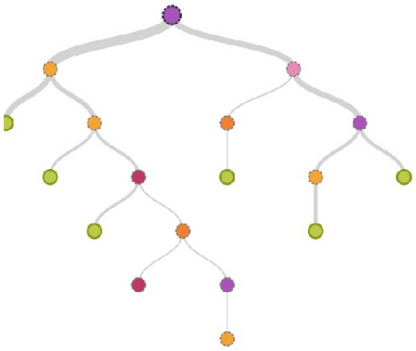
Je pense à une personne. Posez-moi jusqu'à 20 questions oui/non pour déterminer à qui je pense. *Réfléchissez bien à vos questions...*

Comment avez-vous posé les questions ?

Quelle mesure sous-jacente vous a amené les questions, le cas échéant ?

Plus important encore, les questions itératives oui/non de ce type ne nécessitent aucune métrique et sont bien adaptées aux données nominales.





14

20 Questions

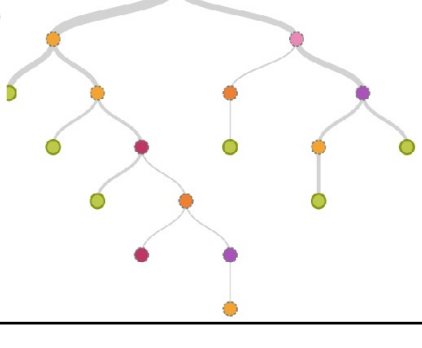
I am thinking of a person. Ask me up to 20 yes/no questions to determine who this person is that I am thinking about.

Consider your questions wisely...

How did you ask the questions?

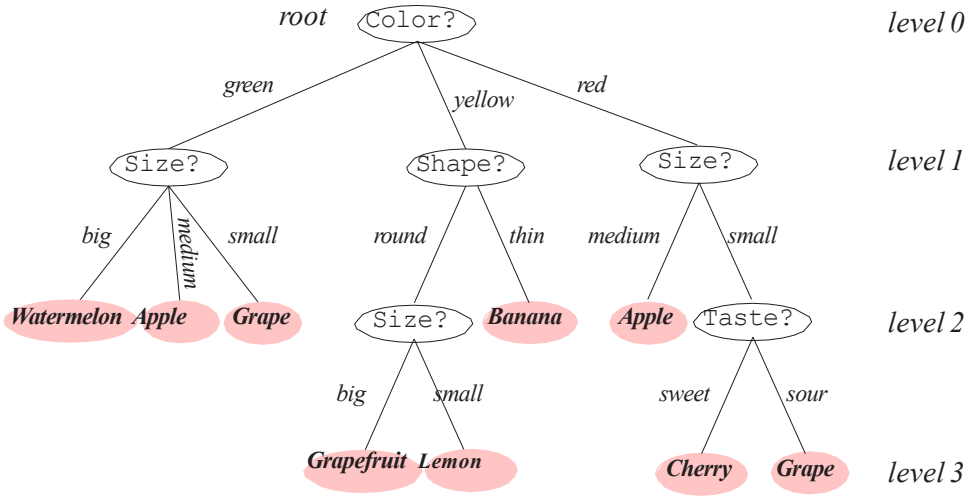
What underlying measure led you the questions, if any?

Most importantly, iterative yes/no questions of this sort require no metric and are well suited for nominal data.



15

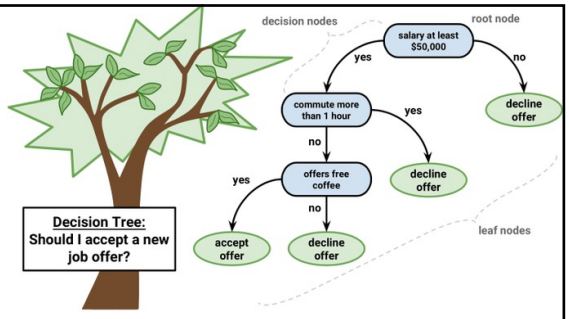
Sequence of questions - a decision tree...



SORBONNE UNIVERSITÉ

16

Decision Trees 101



The **root node** of the tree is connected to successive **branches** to the other **decision nodes**.

The connections continue until the **leaf (terminal) nodes** are reached = a decision.

The classification of a particular pattern begins at the root node, which queries a particular property (selected during tree learning).

The links off of the root node correspond to different possible values of the property.

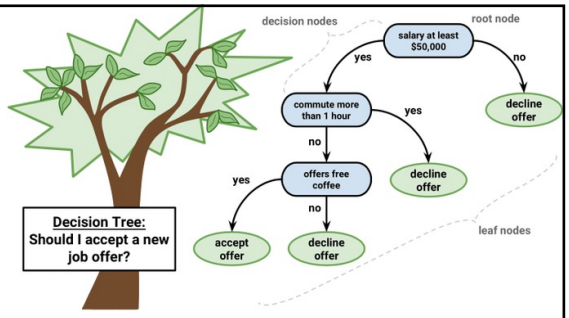
We follow the link corresponding to the appropriate value of the pattern and continue to a new node, at which we check the next property, and so on.

Decision trees have a particularly **high degree of interpretability.**

SORBONNE UNIVERSITÉ

17

Arbres de Décision 101



Le nœud racine de l'arbre est connecté par branches successives aux autres **nœuds de décision**.

Les connexions se poursuivent jusqu'à ce que les nœuds terminaux (feuilles) soient atteints = une décision.

La classification d'un motif particulier commence au nœud racine, qui interroge une propriété particulière (sélectionnée lors de l'apprentissage de l'arbre).

Les liens hors du nœud racine correspondent à différentes valeurs possibles de la propriété.

Nous suivons le lien correspondant à la valeur appropriée du motif et continuons jusqu'à un nouveau nœud, auquel nous vérifions la propriété suivante, et ainsi de suite.

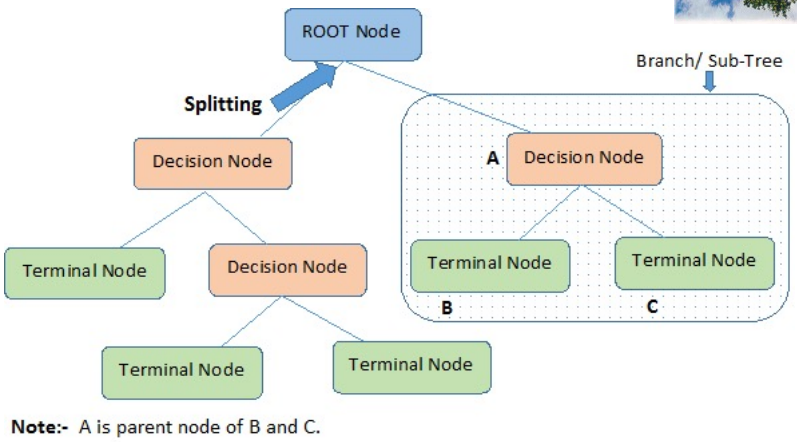
Les arbres de décision ont un **degré d'interprétabilité particulièrement élevé**.

18

SORBONNE UNIVERSITÉ

18

Decision Trees 101



Note:- A is parent node of B and C.

19

SORBONNE UNIVERSITÉ

19

Quand envisager (d'utiliser) les arbres de décision ?
When to Consider Decision Trees?

- Les instances sont entièrement ou partiellement décrites par des paires attribut-valeur.
- La fonction cible est à valeur discrète.
- Une hypothèse disjonctive peut être requise.
- Données d'entraînement potentiellement bruyantes.
- Exemples :
  - ✓ Matériel ou diagnostic médical.
  - ✓ Analyse du risque de crédit.
  - ✓ Modélisation des préférences de planification du calendrier.

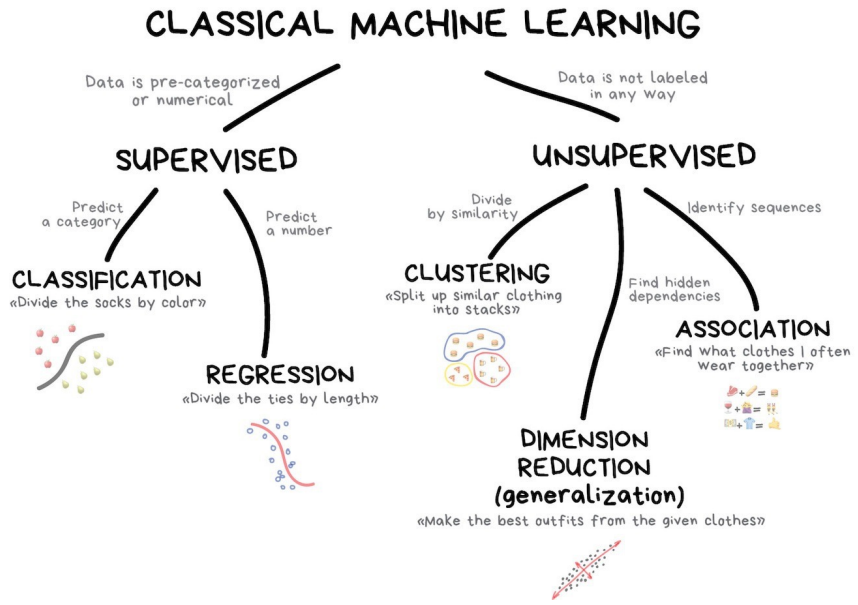
- *Instances are wholly or partly described by attribute-value pairs.*
- *Target function is discrete valued.*
- *Disjunctive hypothesis may be required.*
- *Possibly noisy training data.*
- *Examples*
  - ✓ *Equipment or medical diagnosis.*
  - ✓ *Credit risk analysis.*
  - ✓ *Modeling calendar scheduling preferences.*

20

SORBONNE UNIVERSITÉ

20

Classification vs. Regression : recall …



21

SORBONNE UNIVERSITÉ

21

Apprentissage par arbre de decision (ADD)

Supposons que nous ayons un ensemble de données d'apprentissage étiquetées D et que nous ayons décidé d'un ensemble de propriétés pouvant être utilisées pour discriminer les modèles.

Maintenant, nous voulons apprendre à organiser ces propriétés dans un arbre de décision pour maximiser la précision.

Tout arbre de décision divisera progressivement les données en sous-ensembles.

Si à un moment donné, tous les éléments d'un sous-ensemble particulier sont de la même catégorie, alors nous disons que ce nœud est pur et nous pouvons arrêter la division.

Malheureusement, cela arrive rarement et nous devons décider entre arrêter le fractionnement et accepter une décision imparfaite ou choisir une autre propriété et faire pousser l'arbre davantage.

SORBONNE UNIVERSITÉ

CART for Decision Tree Learning

(CART = Classification And Regression Tree)

Assume we have a set of $\mathcal{D}$ labelled training data and we have decided on a set of properties that can be used to discriminate patterns.

Now, we want to learn how to organize these properties into a decision tree to maximize accuracy.

Any decision tree will progressively split the data into subsets.

If at any point all of the elements of a particular subset are of the same category, then we say this **node is pure** and we can stop splitting.

Unfortunately, this rarely happens and we have to decide between whether to stop splitting and accept an imperfect decision or instead to select another property and grow the tree further.

SORBONNE UNIVERSITÉ

Strategie de definition des ADD

La stratégie de base pour définir de manière récursive un arbre de décision est la suivante : étant donné les données représentées sur un nœud, soit déclarez ce nœud comme une feuille, soit trouvez une autre propriété à utiliser pour diviser les données en sous-ensembles.

Il y a 6 types généraux de questions qui se posent :

- ✓ Combien de branches seront sélectionnées à partir d'un nœud ?
- ✓ Quelle propriété doit être testée sur un nœud ?
- ✓ Quand un nœud doit-il être déclaré feuille ?
- ✓ Comment tailler un arbre une fois qu'il est devenu trop grand ?
- ✓ Si un nœud feuille est impur, comment la catégorie doit-elle être attribuée ?
- ✓ Comment traiter les données manquantes ?

SORBONNE UNIVERSITÉ

The basic CART strategy

The basic CART strategy to recursively defining the tree is the following: *Given the data represented at a node, either declare that node to be a leaf or find another property to use to split the data into subsets.*

There are 6 general kinds of questions that arise:

- ✓ How many branches will be selected from a node?
- ✓ Which property should be tested at a node?
- ✓ When should a node be declared a leaf?
- ✓ How can we prune a tree once it has become too large?
- ✓ If a leaf node is impure, how should the category be assigned?
- ✓ How should missing data be handled?

SORBONNE UNIVERSITÉ

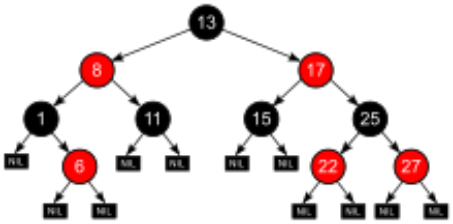
Number of Splits

Le nombre de divisions à un nœud, ou son facteur de ramification B, est généralement fixé par le concepteur (en fonction de la façon dont le test est sélectionné) et peut varier dans l'arbre.

- NB : Tout fractionnement de facteur supérieur à 2 peut facilement être converti en une séquence de fractionnements binaires.

Nous nous concentrons uniquement sur l'apprentissage d'arbres binaires.

- NB : Dans certaines circonstances d'apprentissage/d'inférence, la sélection d'un test à un nœud ou son inférence peut être coûteuse en calcul et une division à 3 ou 4 voies peut être plus souhaitable pour des raisons de calcul.



SORBONNE UNIVERSITÉ

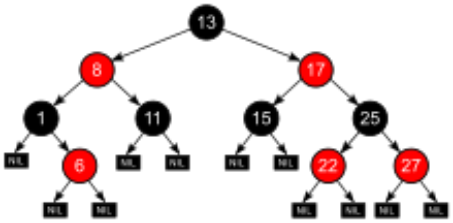
Number of Splits

The number of splits at a node, or its **branching factor** *B*, is generally set *by the designer* (as a function of the way the test is selected) and can vary throughout the tree.

- NB: Any split with a factor greater than 2 can easily be converted into a sequence of binary splits.

We focus on only binary tree learning.

- NB: In certain learning / inference circumstances, the selection of a test at a node or its inference may be computationally expensive and a 3- or 4-way split may be more desirable for computational reasons.



SORBONNE UNIVERSITÉ

Query Selection and Node Impurity

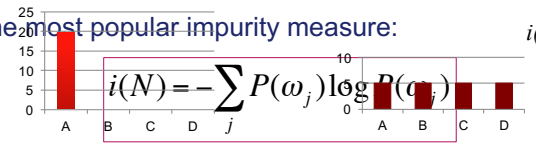
The fundamental principle underlying tree creation is that of **simplicity**: *we prefer decisions that lead to a **simple, compact tree with few nodes**.*

We seek a property query T at each node N that makes the data reaching the immediate descendant nodes as "pure" as possible.

Let $i(N)$ denote the **impurity of a node** N.

In all cases, we want $i(N)$ to be 0 if all of the patterns that reach the node bear the same category, and to be large if the categories are equally represented.

Entropy impurity is the most popular impurity measure: $i(N) = 0$ $i(N)$ ↗



It will be minimized for a node that has elements of only one class (pure).

SORBONNE UNIVERSITÉ

Query Selection and Node Impurity

For the two-category case, a useful definition of impurity is the **variance impurity**:

$$i(N) = P(\omega_1)P(\omega_2)$$

Its generalization to the multi-class is the **Gini impurity**:

$$i(N) = \sum_{i \neq j} P(\omega_i)P(\omega_j) = \frac{1}{2}\left[1 - \sum_j P^2(\omega_j)\right]$$

which is the expected error rate at node N if the category is selected randomly from the class distribution present at the node.

The **impurity misclassification** measures the minimum probability that a training pattern would be misclassified at N:

$$i(N) = 1 - \max_j P(\omega_j)$$

SORBONNE UNIVERSITÉ

Query Selection

Key Question: *Given a partial tree down to node N, what feature s should we choose for the property test T?*

The obvious heuristic is to choose the feature that **yields as big a decrease in the impurity as possible.**

The impurity gradient is: $$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$$

where N_L and N_R are the left and right descendants, respectively, P_L is the fraction of data that will go to the left sub-tree when property T is used.

The strategy is then to choose the feature that maximizes $\Delta i(N)$.

If the **entropy impurity** is used, this corresponds to choosing the feature that yields the **highest information gain**.

Example: PlayTennis ?

New data : Training examples: 9 + / 5 -

Tom M. Mitchell, Machine Learning, McGraw-Hill Science/Engineering/Math, 1997

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

D15	Rain		High	Weak	?

Example: PlayTennis ?

- Hard to guess …
- Try to understand when John plays
- Divide & conquer:
  - Split into subsets
  - Are they pure ? (all yes or all no)
  - If yes: stop
  - If not: repeat
- See which subset new data falls into

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

SORBONNE UNIVERSITÉ

Example: PlayTennis ?

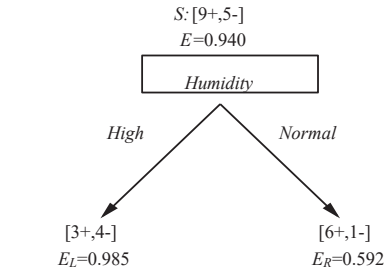
The Root node – entropy (impurity factor) related to the Boolean classification yes/no

$$i(N) = -\sum_j P(\omega_j)\log_2 P(\omega_j) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

SORBONNE UNIVERSITÉ

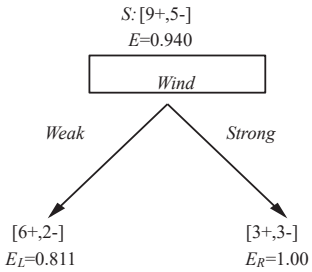
Which attribute is the best classifier?



$E_L = -(3/7)\log_2(3/7) - (4/7)\log_2(3/7) = 0.985$
$E_R = -(6/7)\log_2(6/7) - (1/7)\log_2(1/7) = 0.592$

Gain (S, Humidity)

$= 0.940 - (7/14)\ 0.985 - (7/14)\ 0.592$
$= 0.151$



$E_L = -(6/8)\log_2(6/8) - (2/8)\log_2(2/8) = 0.811$
$E_R = 1$

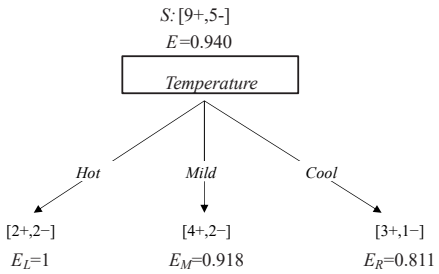
Gain (S, Wind)

$= 0.940 - (8/14)\ 0.811 - (6/14)\ 1.0$
$= 0.048$

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$$ where $$i(N) = -\sum_j P(\omega_j) \log_2 P(\omega_j)$$

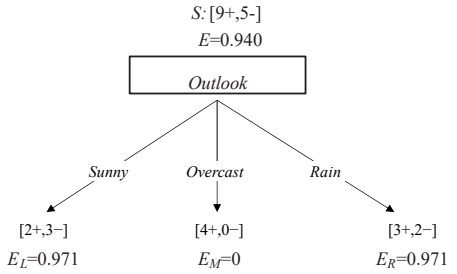
SORBONNE UNIVERSITÉ

Which attribute is the best classifier?



Gain (S, Temperature)

$= 0.940 - (4/14)\ 1.0 - (6/14)\ 0.918 - (4/14)\ 0.811$
$= 0.029$



Gain (S, Outlook)

$= 0.940 - (5/14)\ 0.971 - (4/14)\ 0.0 - (5/14)\ 0.971$
$= 0.246$ – the highest !!

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$$ where $$i(N) = -\sum_j P(\omega_j) \log_2 P(\omega_j)$$

SORBONNE UNIVERSITÉ

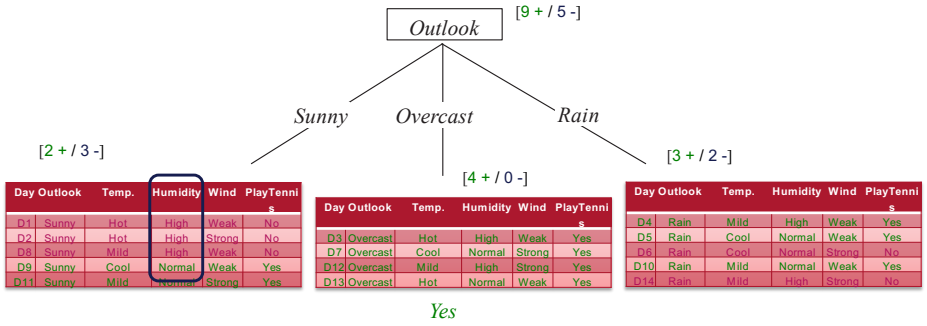
Outlook seems to be the best attribute

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

38

SORBONNE UNIVERSITÉ

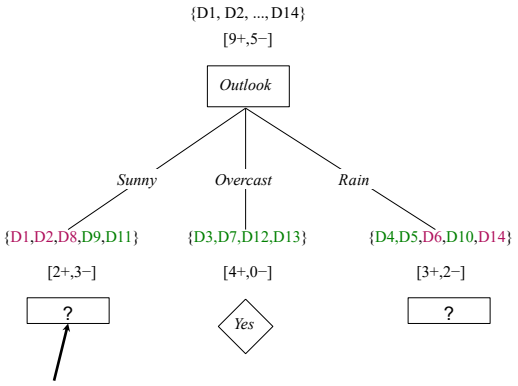
38



39

SORBONNE UNIVERSITÉ

39



Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

Gain $(S_{sunny}$ **, Humidity)** = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain $(S_{sunny}$ *, Temperature)* = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0 = .570

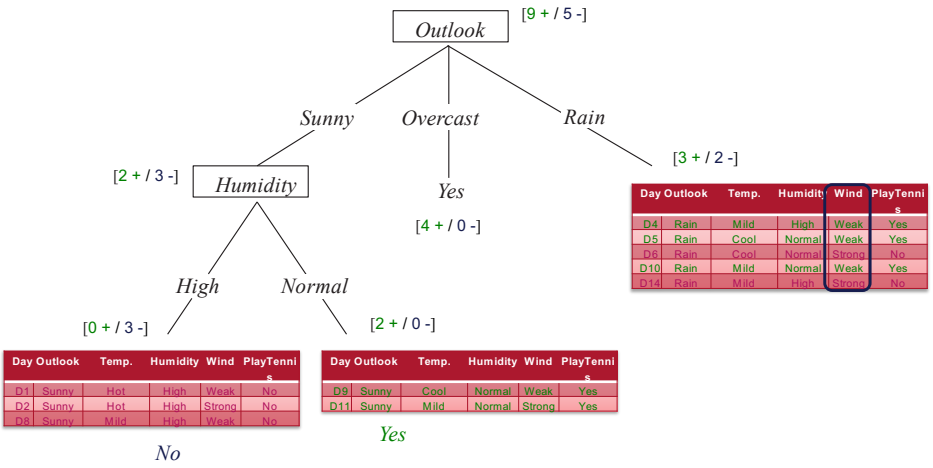
Gain $(S_{sunny}$ *, Wind)* = .970 − (2/5) 1.0 − (3/5) .918 = .019

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$$

where

$$i(N) = -\sum_j P(\omega_j) \log_2 P(\omega_j)$$

SORBONNE UNIVERSITÉ

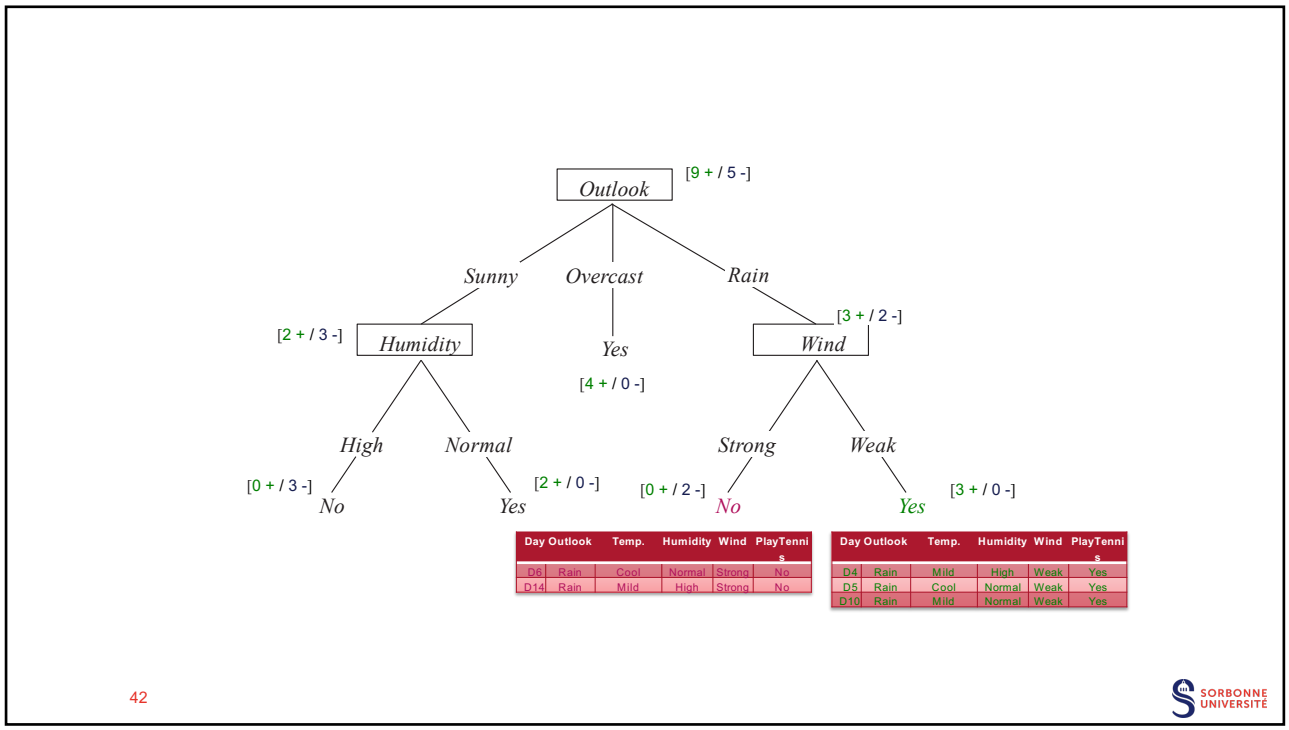


Day	Outlook	Temp.	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No

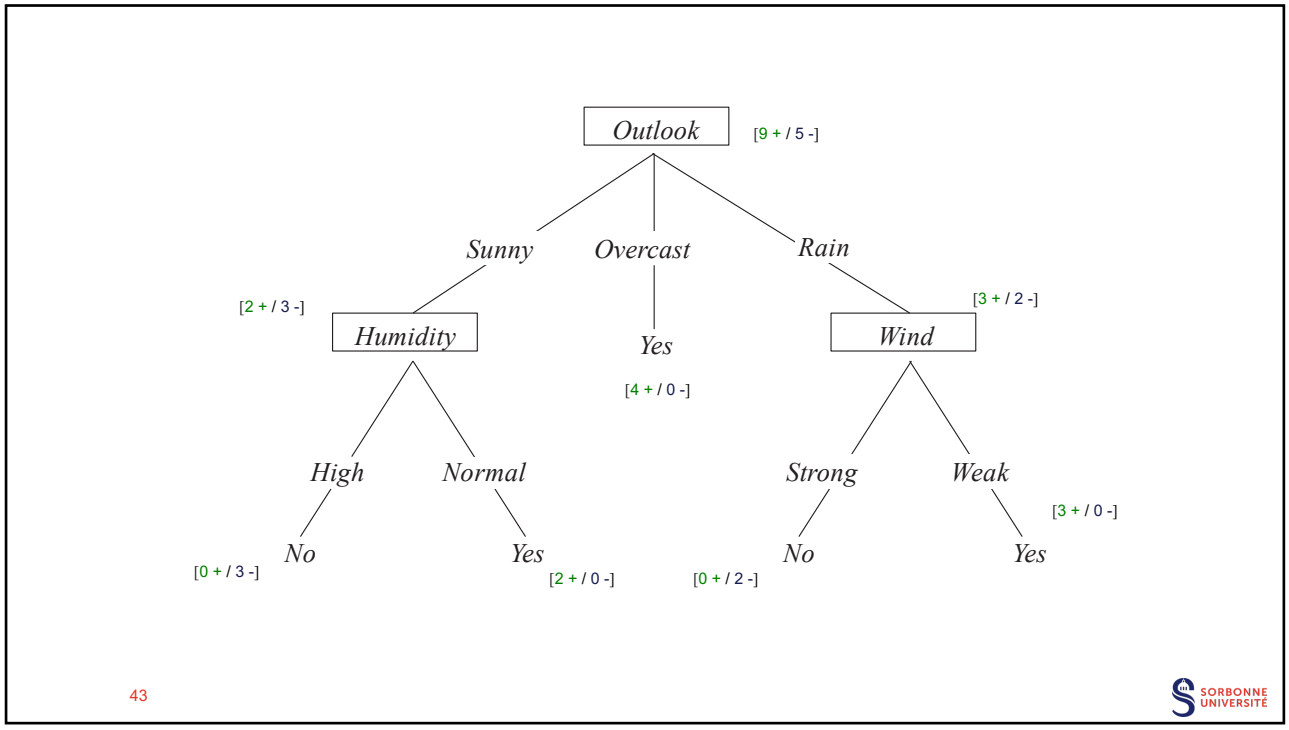
Day	Outlook	Temp.	Humidity	Wind	PlayTennis
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	PlayTennis
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

SORBONNE UNIVERSITÉ


Outlook
[9 + / 5 -]
Sunny
Overcast
Rain
[2 + / 3 -]
Humidity
Yes
[4 + / 0 -]
[3 + / 2 -]
Wind
High
Normal
Strong
Weak
[0 + / 3 -]
No
[2 + / 0 -]
Yes
[0 + / 2 -]
No
Yes
[3 + / 0 -]
Day Outlook Temp. Humidity Wind PlayTennis
D6 Rain Cool Normal Strong No
D14 Rain Mild High Strong No
Day Outlook Temp. Humidity Wind PlayTennis
D4 Rain Mild High Weak Yes
D5 Rain Cool Normal Weak Yes
D10 Rain Mild Normal Weak Yes
42
SORBONNE UNIVERSITÉ

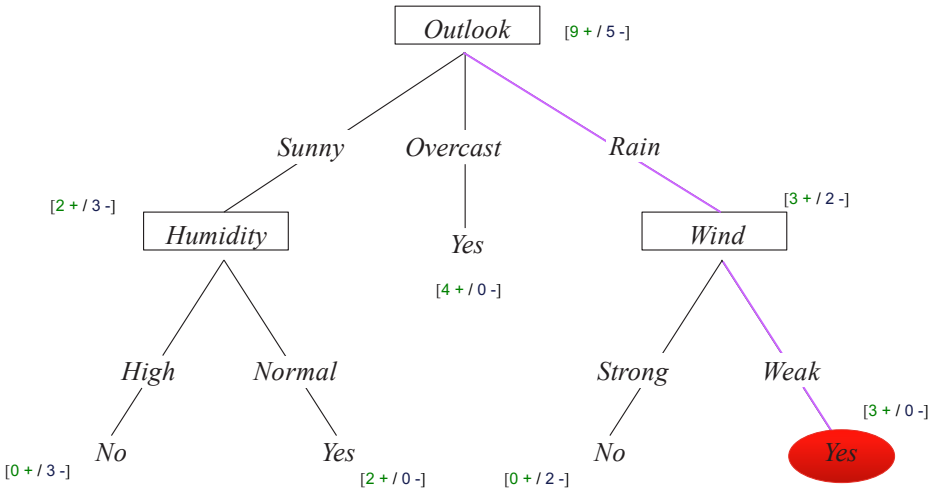
42


Outlook
[9 + / 5 -]
Sunny
Overcast
Rain
[2 + / 3 -]
Humidity
Yes
[4 + / 0 -]
[3 + / 2 -]
Wind
High
Normal
Strong
Weak
[3 + / 0 -]
No
Yes
No
Yes
[0 + / 3 -]
[2 + / 0 -]
[0 + / 2 -]
43
SORBONNE UNIVERSITÉ

43

Learned Tree

Day	Outlook	Temp.	Humidity	Wind	PlayTennis
D15	Rain		High	Weak	Yes



Annexes / Appendix

What can we say about this strategy?

For the binary-case, it yields one-dimensional optimization problem (which may have non-unique optima).

In the higher branching factor case, it would yield a higher-dimensional optimization problem.

In multi-class binary tree creation, we would want to use the **twoing criterion**. The goal is to find the split that best separates groups of the c categories. A candidate "supercategory" C_1 consists of all patterns in some subset of the categories and C_2 has the remainder. When searching for the feature s, we also need to search over possible category groupings.

This is a **local, greedy optimization strategy**.

Hence, there is no guarantee that we have either the global optimum (in classification accuracy) or the smallest tree.
In practice, it has been observed that the particular choice of impurity function rarely affects the final classifier and its accuracy.

SORBONNE UNIVERSITÉ

A Note About Multiway Splits

In the case of selecting a multiway split with branching factor B, the following is the direct generalization of the impurity gradient function:

$$\Delta i(s) = i(N) - \sum_{k=1}^{B} P_k i(N_k)$$

This direct generalization is biased toward higher branching factors.

- ✓ To see this, consider the uniform splitting case.
- ✓ So, we need to normalize each:

$$\Delta i_B(s) = \frac{\Delta i(s)}{-\sum_{k=1}^{B} P_k \log(P_k)}$$

- ✓ And then, we can again choose the feature that maximizes this normalized criterion.

SORBONNE UNIVERSITÉ

When to Stop Splitting?

If we continue to grow the tree until each leaf node has its lowest impurity (just one sample datum), then we will likely have over-trained the data. This tree will most definitely not generalize well.

Conversely, if we stop growing the tree too early, the error on the training data will not be sufficiently low and performance will suffer, again.

So, how to stop splitting?

- ✓ Cross-validation...
- ✓ Threshold on the impurity gradient.
- ✓ Incorporate a tree-complexity term and minimize.
- ✓ Statistical significance of the impurity gradient.

SORBONNE UNIVERSITÉ

Stopping by Thresholding Impurity Gradient

Splitting is stopped if the best candidate split at a node reduces the impurity by less than the pre-set amount, β:

$$\max_s \Delta i(s) \leq \beta$$

Benefit 1: Unlike *cross-validation*, the tree is trained on the complete training data set.

Benefit 2: Leaf nodes can lie in different levels of the tree, which is desirable whenever the complexity of the data varies throughout the range of values.

Drawback: How do we set the value of the threshold β?

SORBONNE UNIVERSITÉ

Stopping with a Complexity Term

Define a new global criterion function

$$\alpha \cdot size + \sum_{left\ nodes} i(N)$$

which trades complexity for accuracy. Here, size could represent the number of nodes or links and α is some positive constant.

The strategy is then to split until a minimum of this global criterion function has been reached.

Given the entropy impurity, this global measure is related to the minimum description length principle.

The sum of the impurities at the leaf nodes is a measure of uncertainty in the training data given the model represented by the tree.

But, again, how do we set the constant α?

50

SORBONNE UNIVERSITÉ

50

… by Testing the Statistical Significance

During construction, estimate the distribution of the impurity gradients Δi for the current collection of nodes.

For any candidate split, estimate if it is statistical different from zero. One possibility is the **chi-squared test**.

More generally, we can consider a hypothesis testing approach to stopping: we seek to determine **whether a candidate split differs significantly from a random split.**

Suppose we have n samples at node N. A particular split s sends P_n patterns to the left branch and $(1 - P)^n$ patterns to the right branch. A random split would place P_{n_1} of the ω_1 samples to the left, P_{n_2} of the ω_2 samples to the left and corresponding amounts to the right.

51

SORBONNE UNIVERSITÉ

51

… Testing the Statistical Significance (cont)

The chi-squared statistic calculates the deviation of a particular split s from this random one:

$$\chi^2 = \sum_{i=1}^{2} \frac{\left(n_{iL} - n_{ie}\right)^2}{n_{ie}}$$

where n_{iL} is the number of ω_i patterns sent to the left under s, and $n_{ie} = P_{ni}$ is the number expected by the random rule.

The larger the chi-squared statistic, the more the candidate split deviates from a random one.

When it is **greater than a critical value** (based on desired significance bounds), we reject the null hypothesis (the random split) and proceed with s.

$$\chi \geq \gamma \; (critical\ value)$$

SORBONNE UNIVERSITÉ

Pruning

Tree construction based on “when to stop splitting” biases the learning algorithm

- ✓ Trees where greatest impurity reduction occurs near the root.
- ✓ No attempt to look ahead at what splits may occur in the leaf and beyond.

Pruning is the principal alternative strategy:

- ✓ In pruning, we exhaustively build the tree. Then, all pairs of neighbouring leafs nodes are considered for elimination.
- ✓ Any pair that yields a satisfactory increase in impurity (a small one) is eliminated and the common ancestor node is declared a leaf.
- ✓ Unbalanced trees often result from this style of pruning/merging.
- ✓ Pruning avoids the “local”-ness of the earlier methods and uses all of the training data, but it does so at added computational cost during the tree construction.

SORBONNE UNIVERSITÉ

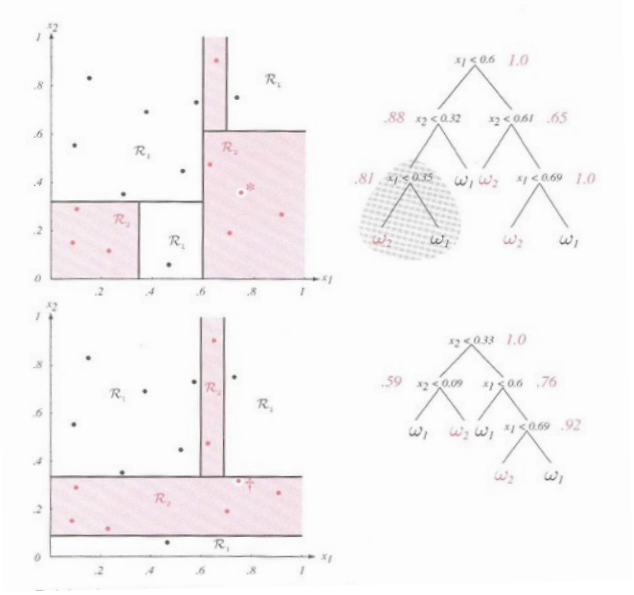
Assignment of Leaf Node Labels and Confidence

A particular leaf node should make the label assignment based on the distribution of samples in it during training. Take the label of the maximally represented class.

Besides, according to the results on the training database, a confidence degree is assigned to each leaf. This will be given as additional result element, in the test phase.

SORBONNE UNIVERSITÉ

Instability of Tree Construction

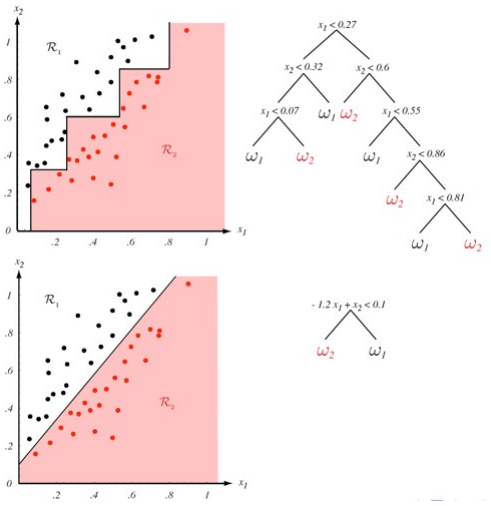


SORBONNE UNIVERSITÉ

Importance of the Feature Choice

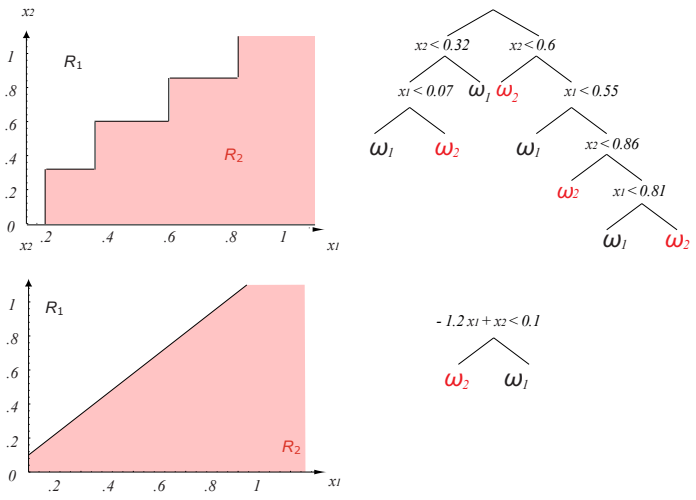
The selection of features will ultimately play a major role in accuracy, generalization, and complexity.

This is an instance of the Ugly Duckling principle.



SORBONNE UNIVERSITÉ

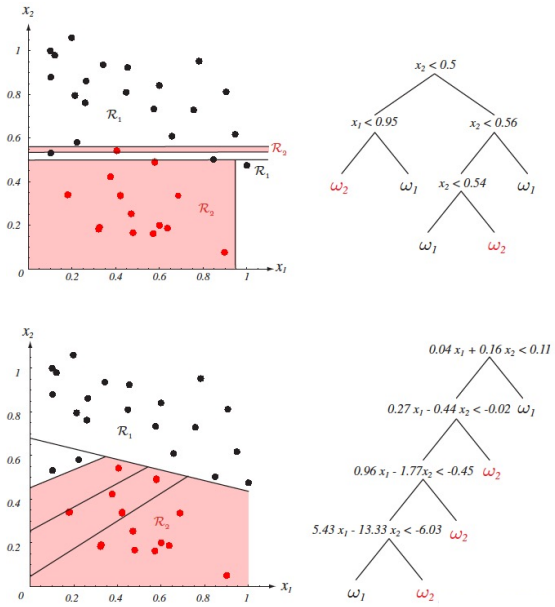
56



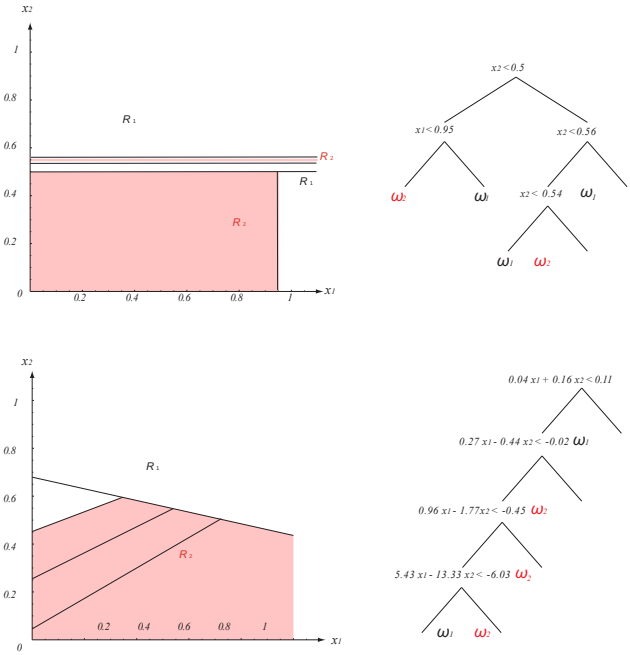
SORBONNE UNIVERSITÉ

57

The use of multiple variables in selecting a decision rule may greatly improve the accuracy and generalization.

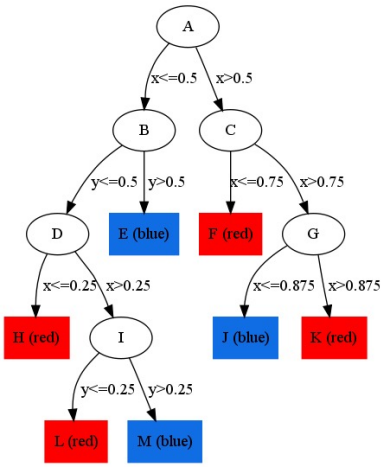


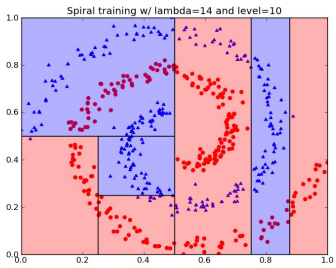
SORBONNE UNIVERSITÉ



SORBONNE UNIVERSITÉ

Dealing with complex problems

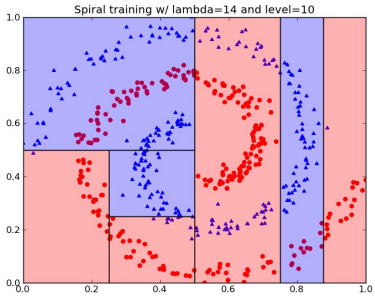


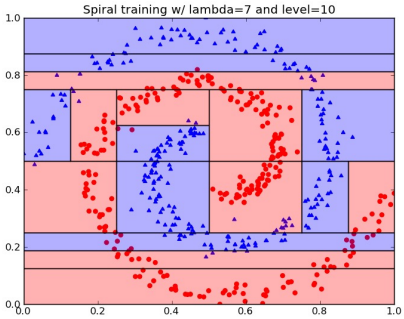


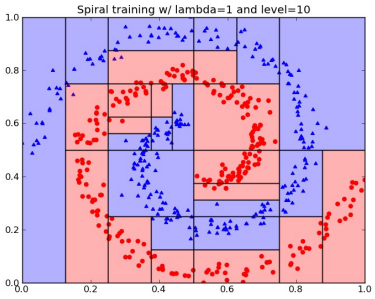
SORBONNE UNIVERSITÉ

60

Dealing with complex problems







SORBONNE UNIVERSITÉ

61

ID3 Method (Quinlan, 1986)

ID3 is another tree growing method.

It assumes nominal inputs.

Every split has a branching factor B_j , where B_j is the number of discrete attribute bins of the variable j chosen for splitting.

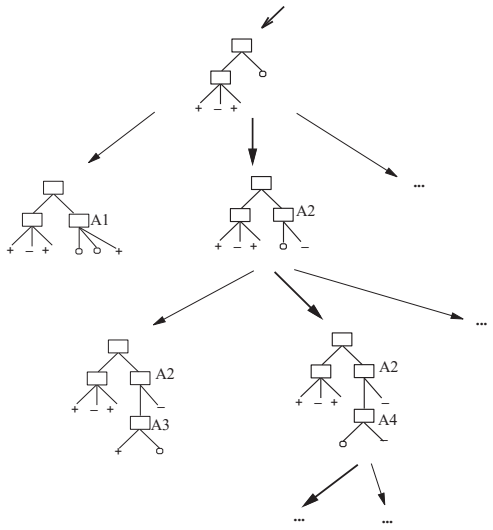
These are, hence, seldom binary.

The number of levels in the trees are equal to the number of input variables.

The algorithm continues until all nodes are pure or there are no more variables on which to split.

One can follow this by pruning.

Hypothesis Space Search by ID3



C4.5 Method (Quinlan, 1993)

This is a successor to the ID3 method.

It handles real valued variables like CART and uses the ID3 multi-way splits for nominal data.

Pruning is performed based on statistical significance tests.

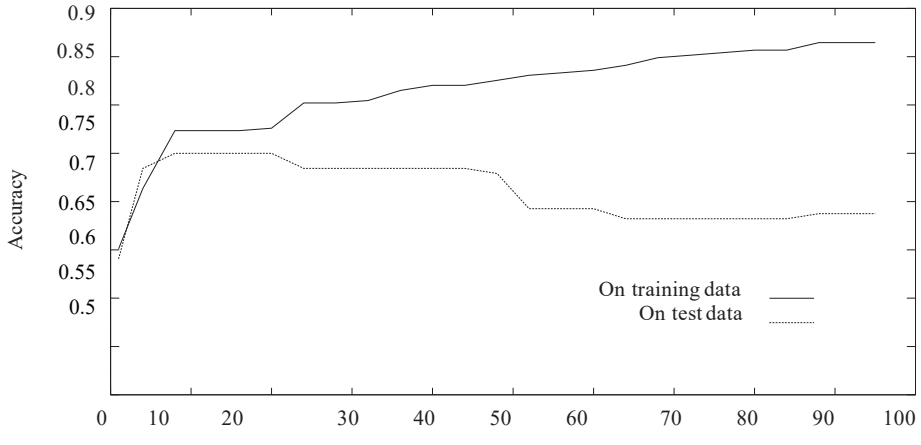
SORBONNE UNIVERSITÉ

Over fitting Instance

Consider adding a new, noisy training example:

Sunny, Hot, Normal, Strong, PlayTennis = No

What effect would it have on the earlier tree?



SORBONNE UNIVERSITÉ